Generating Self-Signed Certificates with VDS-SM

This appendix describes the process for using VDS-SM to generate self-signed certificates that are used with the Splunk process to enable the transaction logs to be transferred from the Cisco Videoscape Distribution Suite, Internet Streamer (VDS-IS) Service Engines (SEs) and Service Routers (SRs) to the VDS-SM using SSL encryption.

Note This appendix provides only one example of generating self-signed certificates. Other tools can be sued to generate these certificates.

This appendix contains the following sections:

Caution PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHORS OF OPENSSL AND CISCO ARE NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFUL, IT IS YOUR RESPONSIBILITY.

The Splunk UF SSL encryption process uses Root certificates, Client certificates, and Server certificates. This appendix provides the steps to generate each of these certificates.

Note When generating the certificates, it is important to follow these guidelines:

- When generating the Root certificate, do not specify a common name or challenge password.
- When generating the Server certificate, do not specify a challenge password.
- When generating the Server certificate, make note of the common name. You will need to reference this common name while configuring SSL on the forwarder.

- When generating the Client certificate, do not specify a common name or challenge password.

Generating a Root Certificate

Follow these steps to generate a Root certificate:

Step 1 Log in to the operating system of any VDS-SM Forwarder.

Step 2 Enter the command **cd /opt/splunkforwarder/bin/** to change directories.

Step 3 To generate the ECDH key that will be used for the Root certificate, enter the following command:

splunk cmd openssl ecparam –name <*curve_name*> **–genkey -noout –out** <*CA_temp_key.***key**>

for example:

```
splunk cmd openssl ecparam –name secp256k1 –genkey –noout –out CATempKey.key
```

Step 4 To encrypt the key that was generated in Step 3 with the AES-256 algorithm, enter the following command:

splunk cmd openssl ec –in <*CA_temp_key.***key**> **–aes256 –out** <*CA_key.***key**>

where <*CA_temp_key.***key**> is the name of the key file generated in Step 3. For example:

```
splunk cmd openssl ec –in CATempKey.key –aes256 –out CAKey.key
```

Step 5 To generate the certificate signing request for the Root certificate, enter the following command:

splunk cmd openssl req -new –[**sha256 | sha384**] **–**key <*CA_key.***key**> **–**out <*csr_name1.***csr**>

where <*CA_key.***key**> is the name of the certificate file generated in Step 4.

Note When prompted, do *not* enter a common name or challenge password.

For example, to generate a certificate signing request that uses SHA-384, enter the following:

```
splunk cmd openssl req –new –sha384 –key CAKey.key –out CACertificate.csr
```

Step 6 To generate the Root certificate, enter the following command:

splunk cmd openssl x509 –req –in <*csr_name1.***csr**> **–**[**sha256 | sha384**] **–signkey** <*CA_key.***key**> **–CAcreateserial –out** <*root_certificate_name.***pem**> **–days** <*valid_time*>

where <*csr_name1.***csr**> is the name of the file created in Step 5, <*CA_key.***key**> is the name of the file created in Step 4, and <*valid_time*> is the number of days for which the root certificate is valid.

For example, to generate a Root certificate that uses SHA-384, enter the following:

```
splunk cmd openssl x509 –req -in CACertificate.csr –sha384 –signkey CAKey.key
–CAcreateserial –out CACertificate.pem –days 1095
```

Step 7 Next, perform the steps in the Generating a Server Certificate section to generate the Server certificates.

Generating a Server Certificate

Follow these steps to generate a Server certificate:

Step 1 From the operating system of any VDS-SM Forwarder, if you are not in the /opt/splunkforwarder/bin/ folder**,** enter the command **cd /opt/splunkforwarder/bin/**.

Step 2 To generate the ECDH key that will be used for the Server certificate, enter the following command:

splunk cmd openssl ecparam –name <*curve_name*> **–genkey –noout –out** <*server_temp_key*.**key**>

for example:

```
splunk cmd openssl ecparam –name secp256k1 –genkey –noout –out ServerTempKey.key
```

Step 3 To encrypt the key that was generated in Step 2 with the AES-256 algorithm, enter the following command:

splunk cmd openssl ec –in <*server_temp_key*.**key**> **–aes256 –out** <*server_key*.**key**>

where <*server_temp_key*.**key**> if the name of the key file that was generated in Step 2. For example:

```
splunk cmd openssl ec –in ServerTempKey.key –aes256 –out ServerKey.key
```

Note The server key name that is created in Step 3 is referenced in the inputs.conf on the VDS-SM.

Step 4 To generate the certificate signing request for the Server certificate, enter the following command:

splunk cmd openssl req –new –[**sha256** | **sha384**] **–**key <*server_key*.**key**> **–out** <*csr_name2*.**csr**>

where <*server_key*.**key**> is the name of the key file that was generated in Step 3.

Note When prompted, do *not* enter a challenge password.

For example, to generate a certificate signing request that uses SHA-384, enter the following:

```
splunk cmd openssl req -new -sha384 -key ServerKey.key -out ServerCertificate.csr
```

Note Make note of the common name that you enter. You will need to reference this common name while configuring SSL on the forwarder.

Step 5 To generate the Server certificate, enter the following command:

splunk cmd openssl x509 –req –in <*csr_name2*.**csr**> **–**[**sha256** | **sha384**] **–CA** <*root_certificate_name*.**pem**> **–Cakey** <*CA_key*.**key**> **–CAcreateserial –out** <*server_temp_certificate*.**pem**> **–days** <*valid_time*>

where <*csr_name2*.**csr**> is the name of the certificate signing request file from Step 4, <*root_certificate_name*.**pem**> is the name of the Root certificate file that was generated in Step 6 of the Generating a Root Certificate section, and <*CA_key*.**key**> is the name of the key file that was generated in Step 4 of the Generating a Root Certificate section. For example, to generate a Server certificate that uses SHA-384, enter the following:

```
splunk cmd openssl x509 –req –in ServerCertificate.csr –sha384 –CA CACertificate.pem
–CAkey CAKey.key –CAcreateserial –out ServerTempCertificate.pem –days 1095
```

Step 6 To consolidate the signed Server certificate, the Server key, and the certificate of the CA into a single PEM file, enter the following command to create a PEM file:

cat <*server_temp_certificate*.**pem**> <*server_key*.**key**> <*root_certificate_name*.**pem**> > <*server_certificate*.**pem**>

where <*server_temp_certificate*.**pem**> is the name of the server certificate file that was created in Step 5, <*server_key*.**key**> is the name of the key file that was generated in Step 3, and <*root_certificate_name*.**pem**> is the name of the Root certificate file that was generated in Step 6 of the Generating a Root Certificate section. For example:

```
cat ServerTempCertificate.pem ServerKey.key CACertificate.pem > ServerCertificate.pem
```

Step 7 Next, perform the steps in the Generating a Client Certificate section to generate the Client certificates.

Generating a Client Certificate

Follow these steps to generate a Client certificate:

Step 1 From the operating system of any VDS-SM Forwarder, if you are not in the /opt/splunkforwarder/bin/ folder**,** enter the command **cd /opt/splunkforwarder/bin/**.

Step 2 To generate the ECDH key that will be used for the Client certificate, enter the following command:

splunk cmd openssl ecparam –name <*curve_name*> **–genkey –noout –out** <*client_temp_key*.**key**>

For example:

```
splunk cmd openssl ecparam -name secp256k1 -genkey -noout -out Client1TempKey.key
```

Step 3 To encrypt the key that was generated in Step 2 with the AES-256 algorithm, enter the following command:

splunk cmd openssl ec –in <*client_temp_key*.**key**> **–aes256 –out** <*client_key*.**key**>

where <*client_temp_key*.**key**> is the name of the key file that was generated in Step 2. For example:

```
splunk cmd openssl ec -in Client1TempKey.key -aes256 -out Client1Key.key
```

Note You will reference the Client key that you generate in Step 5 in the VDS-IS Splunk configuration.

Step 4 To generate the certificate signing request for the Client certificate, enter the following command:

splunk cmd openssl req –new –[sha256 | sha384] –key <*client_key*.**key**> **–out** <*csr_name3*.**csr**>

where <*client_key*.**key**> is the name of the key file that was created in Step 3.

Note When prompted, do not specify challenge password or common name.

For example, to generate a certificate signing request that uses SHA-384, enter the following:

```
splunk cmd openssl req –new –sha384 –key Client1Key.key –out Client1Certificate.csr
```

Step 5 To generate the Client certificate, enter the following command:

splunk cmd openssl x509 –req –in <*csr_name3*.**csr**> **–[sha256 | sha384] –CA** <*root_certificate_name*.**pem**> **–Cakey** <*CA_key*.**key**> **–CAcreateserial –out** <*client_temp_certificate*.**pem**> **–days** <*valid_time*>

where <*csr_name3*.**csr**> is the name of the certificate signing request that was generated in Step 4, <*root_certificate_name*.**pem**> is the name of the Root certificate file that was generated in Step 6 of the Generating a Root Certificate section, and <*CA_key*.**key**> is the name of the key file that was generated in Step 4 of the Generating a Root Certificate section.

For example:

```
splunk cmd openssl x509 -req -in Client1Certificate.csr -sha384 -CA CACertificate.pem
-CAkey CAKey.key -CAcreateserial -out Client1TempCertificate.pem -days 1095
```

Step 6 To consolidate the signed Client certificate, the client key, and the certificate of the CA into a single PEM file, enter the following command to create a PEM file:

cat <*client_temp_certificate*.**pem**> <*client_key*.**key**> <*root_certificate_name*.**pem**> > <*client_certificate*.**pem**>

where <*client_temp_certificate*.**pem**> is the name of the client certificate file that was created in Step 5, <*client_key*.**key**> is the name of the key file that was generated in Step 3, and <*root_certificate_name*.**pem**> is name of the Root certificate file that was generated in Step 6 of the Generating a Root Certificate section. For example:

```
cat Client1TempCertificate.pem Client1Key.key CACertificate.pem > Client1Certificate.pem
```

Next you will install the certificates on the VDS-SM server.

Installing Certificates on VDS-SM

Follow these steps to install the certificates on the VDS-SM server:

Step 1 Enter the following command to copy the certificates to /opt/splunkforwarder/etc/certificates/ directory:

cp <*server_certificate*.**pem**> <*root_certificate_name*.**pem**> **/opt/splunkforwarder/etc/certificates**

where <*server_certificate*.**pem**> is the name of the Server certificate file that was generated in Step 6 of the Generating a Server Certificate section and <*root_certificate_name*.**pem**> is name of the Root certificate file that was generated in Step 6 of the Generating a Root Certificate section.

Step 2 Define the following code stanzas in the inputs.conf file. This file is located in the /opt/splunkforwarder/etc/apps/CDN_UF/local/ folder:

```
[SSL]
rootCA = $SPLUNK_HOME/etc/certificates/<root_certificate_name.pem>
serverCert = $SPLUNK_HOME/etc/certificates/<server_certificate.pem>
password = <server_key_password>
requireClientCert = false
sslVersions = <Required SSL Version>
cipherSuite = <Required ecdh cipherSuite String>
allowSslRenegotiation = true
ecdhCurveName = <ecdh_curve_name>
```

- <*root_certificate_name.pem*> is the name of the Root certificate file that was created in Step 6 of the Generating a Root Certificate section.
- <*server_certificate.pem*> is the name of the Server certificate that was created in Step 6 in the Generating a Server Certificate section.
- <*server_key_password*> is the key that was used to create the Server certificate.
- Set cipherSuite to ECDHE-ECDSA-AES256-GCM-SHA384

Step 3 After you have finished making edits to the inputs.conf file, enter the following commands to restart the Splunk service:

a. **splunk stop**

b. **splunk start**

Validating Configurations

To validate the configurations on VDS-SM, perform the following steps:

Procedure

Step 1 Navigate to the following location: $SPLUNK_HOME/var/log/splunk/splunkd.log

Step 2 In the logs, check for the following lines:

```
INFO TcpInputConfig - IPv4 port 9998 is reserved for splunk 2 splunk (SSL)
INFO TcpInputConfig - IPv4 port 9998 will negotiate new-s2s protocol
```

Note If the SSL connection is wrong, the system raises the following errors:

```
ERROR SSLCommon - Can't read certificate file
/opt/splunk/etc/certificates/ServerCertificate.pem errno=33558530
error:02001002:system library:fopen:No such file or directory
ERROR SSLCommon - Can't read key file
/opt/splunk/etc/certificates/ServerCertificate.pem
```